EARTH SCIENCE MARKUP LANGUAGE: AN UPDATE

Rahul Ramachandran, Helen Conover, Sara Graves, and Ken Keiser
Information Technology and Systems Center
University of Alabama in Huntsville
Huntsville, AL – 35899
rramachandran@itsc.uah.edu

***Abstract*- With continued improvements in technology, the rate of data gathering has grown enormously. However, technology has not been fully exploited in the scientific data analysis process. There are obstacles in data preprocessing that the scientists need to deal with before proceeding with the analysis. Unfortunately in science, there is no single standard data format. Data created by different centers or scientists can be written in numerous formats. The wide variety of data formats leads to the data/application interoperability problem where an application cannot utilize data in different data formats. The solution for the data/application interoperability problem is an interchange technology, such as the Earth Science Markup Language. ESML consists of data descriptions, rules for creating them, and an associated library to provide an elegant solution to this problem. This paper will describe this interchange technology, describe an ESML description file and summarize the future plans for the project.**

I. INTRODUCTION

The wide variety of Earth Science data formats has led to the data/application interoperability problem where an application cannot utilize heterogeneous data sets. Interchange Technology is an enabling technology that utilizes external metadata to allow applications to plug and play seamlessly with datasets in heterogeneous formats. This interchange technology can be utilized to solve the data/application interoperability problem. The Earth Science Markup Language [1,2] is one such interchange technology. Based on XML it consists of an ESML Schema, ESML files and an ESML Library (See Fig 1). ESML files contain descriptions of the content, structure, and semantics of a particular set of data files.

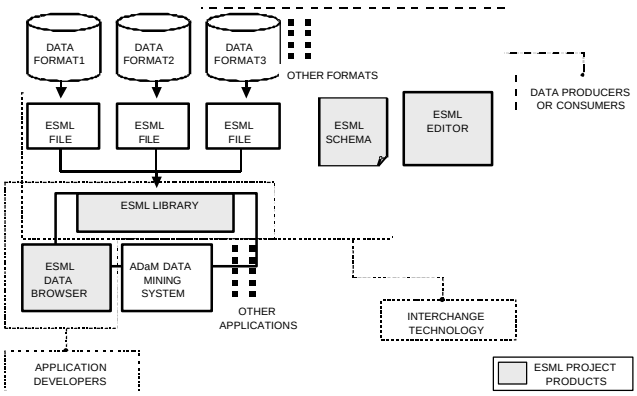


Fig .1. Holistic view of ESML as an interchange technology

The ESML Schema defines rules for creating an ESML file. Because ESML files are external files (i.e. not contained within the data files), both data producers and consumers can create and use them at any time. A key point is that the ESML files do not modify the application or the data file itself. The ESML Library is utilized by applications to parse an ESML file and then they use the information to decode the data format. Application developers can now build data format independent applications utilizing this ESML Library. Furthermore, the application will not require modification in order to access new formats as they become available.

II. ESML SCHEMA

As described earlier, the ESML Schema contains rules for writing valid ESML files. These rules allow users to describe three important aspects of the data file: Content, Structural and Semantic Metadata.

1. Content Metadata

Content metadata describes the dataset in human-readable terms. Although this metadata may be parsed by the ESML reader, it is not necessarily "understood" by the computer. The content metadata documents the origin and pedigree of the data, information that is not typically necessary for automated data manipulation. The information transcribed by content descriptions is usually used for searching and locating information about data sets. For example, content descriptions might tag a data file using element *<discipline>* Atmospheric Chemistry*</discipline>*. This information could be utilized by a student or a scientist searching for data sets under that domain. The design of this schema is mainly based on the EOSDIS Core System (ECS) metadata.

2. Structural Metadata

The structural (or syntactic) metadata can be used to describe the details of the structure of the data stored in a data file in term of bits, bytes and records. All the syntactic metadata in an ESML file are contained within the ESML element tags of *<SyntacticMetaData>* and *</SyntacticMetaData>*. The *SyntacticMetaData* element covers different data format type information, as seen in Fig. 2.

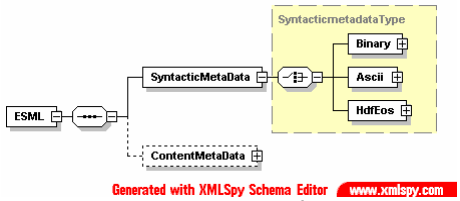


Figure 2: Overall structure of syntactic metadata

The initial formats addressed in ESML are free formats, such as Binary and ASCII text, and structured formats, such as HDF-EOS. The ESML Schema has been designed to be extensible such that other data format elements can be added without affecting the existing design.

3. Semantic Metadata

Semantic metadata is used to give meaning to the elements described by the syntactic metadata. These metadata elements allow the parser to "understand" the actual meaning of the data in terms other than bits and bytes. The semantic metadata is embedded in the syntactic metadata descriptions (within the <*Field*> tags). The ESML Schema provides semantic descriptions for variables in the data file to be tagged as one of *Attribute*, *Time*, *Latitude*, *Longitude*, *Altitude* or *Data*. The basic ESML Schema design goals were not only to allow applications to read the data but also to be able to spatio-temporally navigate the data and utilize the data as actual scientific values. For example, some satellite data brightness temperatures measurements are stored as integers instead of floating point numbers to save space. To correctly utilize these values, proper scaling has to be performed. The ESML Schema provides this capability by allowing a user to specify an equation for data conversion.

III. ESML LIBRARY

The ESML Library provides applications with a programmatic way to read and interpret data files based on the ESML descriptions. A logical data model is used by the ESML Library. Access to data using the ESML Library can be achieved by creating an *ESML_OO* object. This *ESML_OO* contains an array of *ESML_Collections.* An ESML_Collection holds collections of similar data and navigation objects. The ESML_Data object holds all the data fields and methods to access metadata information about the fields. It also provides methods to access the metadata and the data contained within the object. The ESML_Navigation object holds the navigation fields (latitude, longitude, time, etc.) and methods to access both metadata and navigation data contained in it.

IV. ESML UTILITY TOOLS

ESML is packaged with a suite of tools. Users not only have access to the Schema and the ESML Library, they are also provided with an Editor (Fig. 4) and a Data Browser Utility (Fig. 5).

1. ESML Editor

For scientists not familiar with XML, writing ESML tags can potentially be as complicated as writing a new data decoder. To alleviate this problem, an intelligent, easy to use Editor has been created. This ESML Editor provides scientists a user-friendly interface to create an ESML file without worrying about the underlying XML tags.

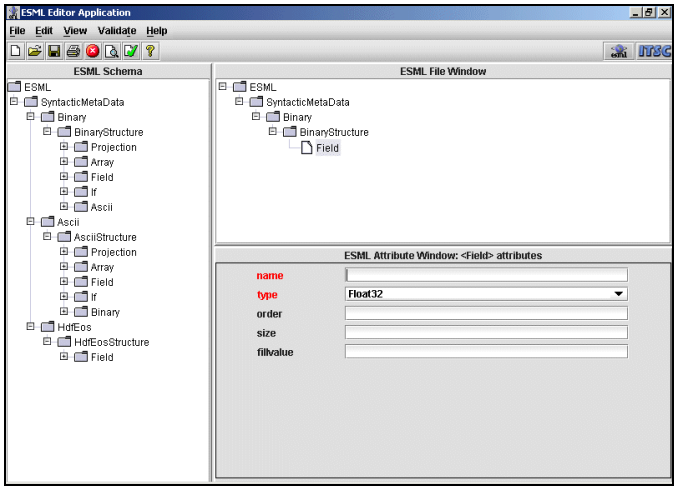


Figure 4: ESML Editor Tool

2. ESML Data Browser

The ESML Data Browser is an example of an application that utilizes the ESML Library. This application has been designed for scientists to give them the ability to view any data file that has an ESML description. This tool can also be used by scientists to test whether the ESML descriptions they have created for datasets are indeed correct.

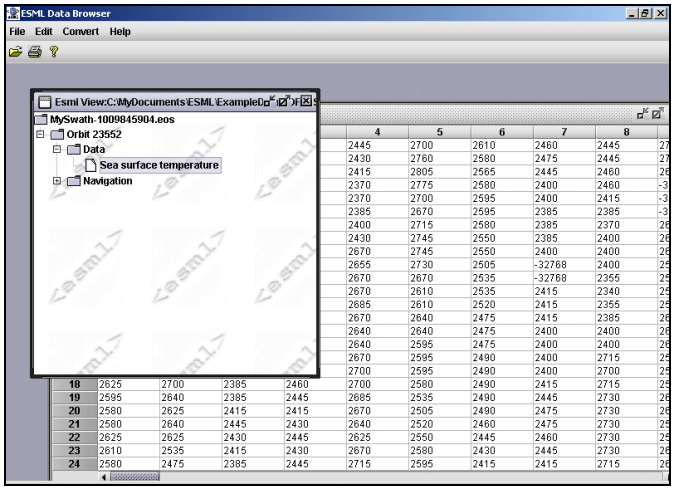
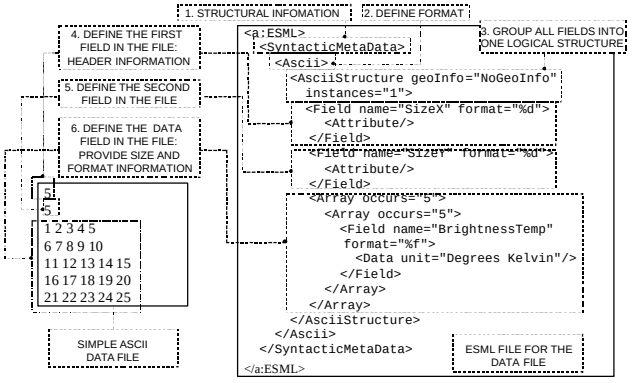


Figure 5: ESML Data Browser Tool

V. AN EXAMPLE ESML DESCRIPTION FILE

An ESML description for a simulated data file in ASCII format is depicted in Fig. 3. The data file consists of two header fields and a single two dimensional data field. The ESML file begins by specifying that this description is



regarding the structure of the data file (3.1).

Figure 3: An example ESML description for simple data set

This is followed by a description of the format of the data file, which in this case is ASCII (3.2). The entire file is grouped in a single logical structure (3.3) with no navigation information. The first two *Fields* are then specified with their names and format types (3.4, 3.5). These two *Fields* are subsequently tagged as *Attributes*. This instructs the ESML parser to ignore these fields and skip to the next *Field*. The next description is the data field (3.6). The two dimensional array is specified by nesting *Array* elements and setting the dimensions to *occurs* in the array element. The *Field* is also nested within these array elements, specified with a name and format. This *Field* is tagged as *Data* with units of the set as "Degrees Kelvin". This semantic tag instructs the parser to read these fields and return any values requested by the user.

VI. ESML ADVANTAGES

ESML is a solution to the data/application interoperability problem that has long existed in the science community. ESML provides an elegant solution to this problem, so that a scientist no longer has to worry about data formats. ESML will allow scientists to utilize a wide variety of data formats and yet allow interoperability with different applications. ESML is not a new data format. An ESML file can be viewed as a "smart" README file. Both the scientists and the applications can understand this file and use it for extracting data regardless of format. Since ESML descriptions are external files, both the data producers and the data consumers can write them. The scientist has only to implement a single reader or data decoder for their analysis or visualization application utilizing the ESML Library. This single reader will then be able to read and understand a wide variety of data formats utilizing the ESML markup files. The scientist can also manipulate the data decoder or the data reader using the ESML tags. Based on the tags, the scientist can make the application slice or dice the data depending upon the problem requirements. The content metadata of ESML allows scientists to describe their data sets and publish this description on the web, to provide mechanisms for data sharing, thereby promoting research. Another nice feature about ESML is that it provides basic preprocessing capabilities through its equation functionality. ESML allows an application to use legacy data sets without costly data format conversion. A single ESML file can be used to describe multiple data files having identical structural and semantic information.

VII. SUMMARY

The combination of the ESML files, schema and the ESML Library-enabled applications form the new interchange technology that will allow applications to switch between different data sets seamlessly. The ESML project has garnered enormous interest from both the science and the application developer community. There are currently several projects leveraging the ESML interchange technology to address the data/application interoperability issue and the metadata description standards. In addition to collaborating with these projects, the ESML team is planning to add new features to the current released version. The three main features that will be added shortly are URL data access for the library, a LINUX version of the ESML Library, and the addition of data formats (GRIB and McIDAS) to both the schema and the library.

Additional information about ESML and its products can be found at http://esml.itsc.uah.edu.

ACKNOWLEDGEMENTS

This research work has been funded by NASA's Earth Science Technology Office. The authors would like to acknowledge Bruce Beaumont, Andrew McDowell, Matt Smith, Xiang Li and Sunil Movva for their contribution to this research.

REFERENCES

[1] R. Ramachandran, M. Alshayeb, B. Beaumont, H. Conover, S. Graves, X. Li, S. Movva, A. McDowell and M. Smith: "Earth Science Markup Language: A Solution for Generic Access to Heterogeneous Data Sets,", Earth Science Technology Conference, Maryland, August 28-30, 2001

[2] R. Ramachandran, M. Alshayeb, B. Beaumont, H. Conover, S. Graves, N. Hanish, X. Li, S. Movva, A. McDowell and M. Smith: "Earth Science Markup Language", 17th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, 81st AMS Annual Meeting, Albuquerque, NM, January, 2001